

Tib: a Reference Setting Package for T_EX

J. C. Alexander
University of Maryland

The purpose of this note is to advertise a preprocessor for T_EX, called Tib, which creates and sets citations and reference lists.

History

Much of my writing is research papers to be published in professional journals. Each such paper requires a reference list. It was clear to me with the first paper I set in T_EX that the error-prone drudgery of creating reference lists should be automated as much as possible. Moreover, I felt, based on several experiences, that any program which created the reference list should be able to set it in a variety of styles and formats (and concomitantly, that users should be able to design new styles). The final impetus was provided by an editor to whom I had submitted a camera-ready manuscript. He responded that the paper was fine, but the publisher specified a different format for the reference list. I reset it, and immediately asked around about bibliography setters on our system, a mainframe running under Unix.

At the time, BIBT_EX was known about, but as something that might be available 'someday,' presumably as part of some future version of L^AT_EX. The *troff* program *refer* was available, and there was a successor to *refer*, called *bib*, written by T. A. Budd, which had more features than *refer* and gave the user more control over the output. I decided to spend a week or so modifying *bib*'s output statements to produce T_EX output (it would also be a good opportunity for a FORTRAN programmer to learn C). Instead of course, I started modifying features and adding new ones, so that the program evolved over several months into something quite different, with a new name, although retaining the basic ideas of *bib*. A number of colleagues, both on our system and others, expressed interest in the program and I eventually took time to write some documentation. It has been in general use since the beginning of the year, mostly for writing scientific papers and theses, and seems to be useful enough that others in the T_EX community might like to try it.

Use

Tib is a preprocessor and does not function as part of a T_EX run. Its use can be described schematically as follows:

$$\text{reference file} \xrightarrow{\text{tibdex}} \text{index,}$$

$$\text{inputfile.tex} + \text{index} \xrightarrow{\text{tib}} \text{outputfile.tex.}$$

The reference file is a data base of bibliographic reference items in the format of a *refer* reference file. The index is a sorted list of all words in the reference file. It can be made once and kept. Using the index, Tib can quickly track down a reference item from one or two words. With the use of indices, Tib can handle large reference files. The input file is a T_EX file with special symbols where citations are wanted and another symbol where the reference list is wanted. The citation symbols surround words from the reference list such as authors' names or a word from the title. When Tib encounters one of these, it 'looks up' the reference, sets the citation, and later processes the reference items and sets the reference list at the indicated location. The output file is another T_EX file with control strings and `\include` statements to completely set the citations and reference list. The special symbols do not interfere with T_EX processing, so T_EX can be applied to the input document for preliminary proofing and debugging. If necessary, the output file can be edited, for example to eliminate an overfull hbox. Each of the arrows above can be many-to-one. That is, one index can be built from several reference files, and several indices and several input files can be used to create one output file.

There are two other programs in the Tib package besides *tib* and *tibdex*: *tiblist*, which sets all the references in a reference list (e.g. for a *vita*), and *tiblook*, which permits the user to interactively look at entries in a reference file.

Tib is designed to be easy to use. There are some demonstrations provided with the package and emulating them is the best way to get started. On the other hand, Tib is built so that a user can customize it or design an individual style with a straightforward effort.

Features

Some of the features of Tib:

1. Tib is not dependent on any particular macro package. It works with plain T_EX files, with L^AT_EX files, with AMS-T_EX files. I have not yet tried T_EX preprocessors such as Larry Siebenmann's Sweet-T_EX (AMS *Notices*, August, 1986), but there should be no conflicts between Tib and such preprocessors.
2. There are two sets of citation symbols, for setting citations in running text or not, and text can be set within a citation.
3. Tib will set citations and reference lists in a variety of styles. For example, citations can be set as reference numbers ('[1]' or '(1)'), as alphabetic codes based on the authors' initials (with or without date), or as last names (also with or without date). Included in the Tib package are several basic styles, including footnotes, as well as the styles of various journals and societies. Included are three AMS styles, two SIAM styles, and ones for IEEE, ACM, APS and AGU journals. A default style is provided, or the user can select the style by a single flag on the program call.
4. Styles are created in a modular fashion, and new ones can be made in a straightforward manner. A user can create his or her own style. In particular, a journal could accept an electronic manuscript processed by Tib and set the manuscript with its own style macros. Of course, the author should have previewed the paper using those macros.
5. Most styles call for processing reference items. Tib can abbreviate first names, reverse first and last names, put names in caps-small caps (if such a font is available) and sort. Such things as T_EX's diacritical marks are correctly handled. Tib can handle some subtleties of English usage. Alphabetizing is done by the first capital letter of the last name. For example, 'deRham' is placed in the R's. Abbreviation is done with some care. 'Heinz-Otto' is abbreviated 'H-O.' 'd'Arcy' is abbreviated 'd'A.'
6. Tib will set up and use 'word-definition' pairs. For example, depending on the style, the 'word' '[UNIV]' in a reference item can be expanded to 'Univ.' or 'University' (or anything else). In particular, *Mathematical Reviews* has provided its complete set of over 1800 journal abbreviations, and Tib can automatically provide the correct abbreviation for these journals. A staff member of my university's library told me

that over 60% of inquiries at the information desk concerned incorrect journal abbreviations. Also a list of publishers is provided, set up so that the city of publication is automatically included.

7. The computer's environment can be used to tailor Tib. For example, Tib searches a default index if it cannot find a reference otherwise. The default is a system index, but the user can change this in the environment. Thus for example, the user could list all personal publications in a special file, make an index for it, and it will always be available for Tib. Similarly, the user can change the default style in the environment. If mostly the ACM style is used, it can be made the user's default.
8. A wider variety of fields can be handled than with the original *refer* data bases. For example, for a translated item, both the original and translated versions can be included in one reference item.

Example

This example demonstrates basic Tib use. The following might be an entry in a reference file:

```
%A F. R. Gantmacher
%A M. G. Krein
%I State Press for Technical Literature
%C Moscow-Leningrad
%T Oscillation Matrices and Kernels and
Small Vibrations of Mechanical Systems
%o (Russian)
%D 1950
%t Oszillationmatrizen, Oszillationskerne
und Kleine Schwingungen Mechanischer Systeme
%a Alfred St{"o}hr
%i Akademie-Verlag
%c Berlin
%d 1960
%O German translation
```

In the input T_EX file, this reference might be cited as follows:

```
... the theory of Jacobi
matrices.[.krein small.] Furthermore ...
```

Tib is applied to this file, and T_EX to the resulting file. In the final printed document, with one of the AMS styles, the line above appears as:

```
... the theory of Jacobi matrices [Gantmacher
and Krein, 1950]. Furthermore ...
```

The corresponding entry in the reference list is:

F. R. Gantmacher and M. G. Krein [1950], *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems* (Russian), State Press for Technical Literature, Moscow-Leningrad, *Oszillationmatrizen, Oszillationskerne und Kleine Schwingungen Mechanischer Systeme*, Akademie-Verlag, Berlin, 1960, German translation.

Portability

The source language of Tib is standard C. It was written on a VAX mainframe operating under Unix, but some attention has been paid to portability. A day was spent with Nelson Beebe at the University of Utah going over changes necessary to bring it up on a non-Unix system. These changes mostly consisted of simplifying file names (over the objections of some Unix devotees), and isolating and documenting system-dependent parts (such as a file sort). Although I have not tried it, I see no particular reason why Tib should not work on a micro. The programs are not particularly big. It might be that large 'word-definition' files should not be used on a micro; the time Tib consumes setting them up might be prohibitive.

Availability

The Tib package of source files, macros and documentation consists of 85 files using about 640K. Included are step-by-step instructions for installation, a list of things non-Unix users should note, a plain TeX source file for a 22 page manual, and some demonstration and test files. The package has been given to Rick Furuta for inclusion with the Unix TeX distribution. It is also available for transporting via *ftp* from *eneevax.umd.edu*. Login anonymously, change to the subdirectory *pub/tib*, and copy everything. A third possibility is to get in touch with me about copying a tape. If there is enough interest that it is worth making future versions, they will be announced in the *TUGboat*.

Packed File Format Update

Tomas Rokicki

Some errors in my description of the packed file format and a bug in one of the conversion programs have been brought to my attention in the past year by John Crawford, Wayne Sewell, and others. This note addresses these problems and lists the changes made to the programs. The new, updated programs are on the current distribution tape; they are also available via anonymous FTP from *SU-Score.EDU* in the directory `<TEX.MF>`.

The first clarification concerns the number of no-op bytes allowed at the end of a packed file. In my original description, I stated that the postamble command was 'followed by just enough no-op commands to make the file a multiple of four bytes long.' For those machines which write binary files in larger blocks, this is inconvenient; thus, I have changed this to allow any number, including zero, of no-op bytes at the end. I have also removed a check for the number of no-op bytes from `pktype`.

The second problem relates to the design size; in the description of the format, I state that the units are $1/2^{16}$ points. Actually, they are in the same units as in the generic font and font metric files: FIXes, or $1/2^{20}$ points.

Finally, in `gftopk`, the minimum bounding box of a character glyph was being calculated incorrectly on the right side. Occasionally an extra row of blank pixels was left in the character. This should not adversely affect any programs; the TFM width and escapement values are still correct, as is the positioning of the reference point. The only manifestation of the problem is in slightly (less than a percent, usually) larger packed files than are absolutely necessary.

All further corrections as listed below pertain to the new version numbers and the change from the old Almost Modern fonts to the new Computer Modern fonts. The following changes should be made in the WEB sources rather than the change files, as they are updates and not system dependent fixes.